Ejercicio para la 2ª sesión de laboratorio

Se pide construir un programa que muestre por orden de fecha la información de los alquileres de una agencia de alquiler de coches. Se dispone de dos archivos, uno (coches.txt) con la información de códigos (enteros) y nombres (cadenas) de los coches de los que se dispone. El archivo empieza con un entero indicando el número de coches y continua con el código, precio por día y nombre de cada coche (ordenados de menor a mayor código), separados por espacios y saltos de línea cómo se muestra a continuación:

```
10
1620 30 Ford B-Max
1621 38 Ford C-Max
1722 32 Toyota Auris
...
```

El otro archivo (rent.txt) contiene (sin ningún orden) la relación de alquileres que se han contratado (código del coche, fecha en formato AA/MM/DD y días que se ha alquilado). El fichero también comienza con el número de alquileres:

```
8
1722 13/3/17 1
1620 1/2/15 7
1722 15/7/18 6
7777 1/1/18 5
...
```

El programa deberá empezar cargando la información de cada archivo en las estructuras `ListaCoches` y `ListaAlquileres`, ambas formadas por un array dinámico (de elementos de tipo `Coche` y `Alquiler` respectivamente), su tamaño y un contador indicando el número real elementos (el tamaño debe ser algo mayor que el número real de elementos, por ejemplo diez más). La lista de coches quedará ordenada por orden de códigos (como en el archivo). Una vez leídas las listas ordenará la lista de alquileres por fechas. El programa terminará mostrando la información sobre los alquileres (fecha, modelo, días alquilado y precio) como se indica:

```
1/2/15 Ford B-Max 7 dia(s) por 210 euros
13/3/17 Toyota Auris 1 dia(s) por 32 euros
1/1/18 ERROR: Modelo inexistente
15/7/18 Toyota Auris 6 dia(s) por 192 euros
...
```

Recuerda borrar la memoria dinámica creada. El programa deberá hacer uso de las siguientes funciones:

- cargarCoches: carga la información del archivo coches.txt en la lista de coches; devuelve true si se ha podido abrir el archivo y false en caso contrario. La lista de coches sólo contendrá la información de este archivo.
- leerAlquileres: carga la información del archivo rent.txt en la lista de alquileres; devuelve true si se ha podido abrir el archivo y false en caso contrario. Cada alquiler debe incluir un campo de tipo `Coche*` con el puntero al coche al que hace referencia el alquiler (que será `nullptr` en caso de no encontrarse el código en la lista de coches). Para obtener dicho puntero deberás llamar a la función buscarCoche (ver abajo).
- ordenarAlquileres: ordena la lista de alquileres por orden de fecha (menor a mayor). Para ello puedes utilizar la función `sort` de la librería `algorithm`, la cual requiere tres parámetros: un puntero al comienzo del array, un puntero al primer elemento fuera de él y el nombre de la función que implemente la operación de orden *menor-que* entre alquileres.
- buscarCoche: dada la lista de coches y un código, devuelve un puntero al coche de la lista con ese código o el puntero `nullptr` si no se encuentra.
- mostrarAlquileres: dada la lista de alquileres, muestra la relación de alquileres con el formato mostrado arriba.

Dado que aún no hemos visto ni módulos ni clases, puedes implementar todo (definiciones de tipos y funciones) en el mismo fichero fuente .cpp. Eso sí, ten en cuenta que el compilador procesa el fichero de arriba a abajo y por lo tanto no puedes usar tipos ni funciones si no se han definido más arriba en el fichero. En el CV puedes encontrar un ejemplo de ficheros de entrada y un fichero (salida.txt) con la salida que tu programa debería generar para dichos ficheros de entrada.

Sobre la entrega: Se debe realizar una entrega por alumno en la cual solo debéis subir vuestro fichero fuente .cpp. La entrega se realiza en el CV, en la pestaña Prácticas.

Ejercicios adicionales: 1) Modifica la estructura (y actualiza todo lo necesario) de manera que la estructura `ListaCoches` se represente mediante un array dinámico de punteros a estructuras de tipo `Coche`. 2) Reorganiza el código usando orientación a objetos y módulos.